FlexSplat: Flexible Feed-Forward 3D Gaussian Splatting without Point Cloud Correspondence

Amir Sabbaghziarani[1,2]
asabbaghziarani1@gsu.edu
asabbagh7@gatech.edu
Hanting Ye[3]
hanting.ye@duke.edu
Maria Gorlatova[3]
maria.gorlatova@duke.edu
Yi Ding[4]
yding@utk.edu

[1] Tri-Institutional Georgia Institute of Technology/Georgia State University/Emory University Center for Translational Research in Data Science and Neuroimaging (TReNDS) Atlanta, GA, USA
[2] Georgia State University Atlanta, GA, USA
[3] Duke University Durham, NC, USA
[4] University of Tennessee, Knoxville Knoxville, TN, USA

Abstract

We present FlexSplat, a feed-forward framework for novel view synthesis (NVS) from *uncalibrated*, object-centric multi-view image collections. A recent line of query-based methods reconstructs a compact set of 3D Gaussians by treating them as transformer queries that are refined with multi-view deformable attention; these methods, however, assume that camera poses are given. FlexSplat removes this assumption: a geometry transformer is trained *jointly* with the Gaussian decoder to predict per-image camera parameters and depth, which in turn ground a depth-guided Gaussian parameterization and a multi-view deformable cross-attention that aggregates evidence across all input views into a single, view-consistent set of primitives. An uncertainty-weighted depth-consistency objective lets the jointly trained geometry adapt to the reconstruction task, while the cross-view consensus formed during decoding absorbs the residual error of the estimated cameras and depth. The representation uses a compact Gaussian budget that is decoupled from the input resolution — unlike pixel-aligned methods, the primitive count does not grow with the image grid — and is not dictated by the number of views. On ShapeNet-SRN and Google Scanned Objects (GSO), FlexSplat matches or approaches posed state-of-the-art reconstructors while requiring *neither* camera poses *nor* ground-truth depth, and matches the best perceptual (LPIPS) quality among the compared methods on GSO. Our results indicate that a jointly trained geometry front-end is sufficient to bring calibration-free operation to query-based Gaussian reconstruction while staying within 0.7 dB PSNR of posed methods and matching their perceptual quality. The code is available at https://github.com/amir-sbg/FlexSplat.

© 2026. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

1 Introduction

Novel view synthesis (NVS) aims to render a scene from unseen viewpoints given a set of input images. Neural Radiance Fields (NeRF) [15] and its variants [2, 16] achieve striking photorealism, and 3D Gaussian Splatting (3D-GS) [12] adds real-time rendering through explicit anisotropic primitives. Both, however, require accurate camera poses from Structure-from-Motion [19] and per-scene optimization that takes minutes to hours. Feed-forward reconstructors [3, 4, 23] remove the per-scene optimization by predicting a 3D representation directly from images, but most of them still place Gaussians through an *explicit correspondence*. Two correspondence paradigms dominate. In the *pixel-aligned* paradigm [3, 20, 23], each pixel of each input view is mapped to one Gaussian, so the number of primitives grows with the image resolution and the number of views. In the *point-based* paradigm [33, 42], Gaussians are anchored to dense 3D points or triplane features predicted by a multi-stage geometry pipeline. We use these two terms throughout in this precise sense: a pixel-aligned primitive is tied to an image location, a point-based primitive is tied to a predicted 3D location.

Both paradigms share a structural cost. Tying primitives to pixels or points forces the model to spend Gaussians on the input parameterization rather than on the object, producing redundant, overlapping primitives that overfit appearance, inflate memory, and blur regions where views disagree [29, 30]. A recent and effective alternative breaks the correspondence entirely: Gaussians are modelled as *learnable transformer queries*, each query is a 3D ellipsoid whose centre is a reference point projected onto image features, and a deformable decoder refines the queries layer by layer. LeanGaussian [29] introduced this idea for the *single-view* case, and UniGS [30] extended it to multiple *posed* views through a multi-view deformable cross-attention that aggregates a single, "unitary" set of Gaussians across views. These methods produce compact, high-quality reconstructions, but they all assume that camera poses (and often depth) are available at input time — precisely the preprocessing that feed-forward NVS set out to avoid.

FlexSplat closes this gap for the object-centric setting. Our scope is deliberate: we target uncalibrated reconstruction of *objects* — the regime of asset creation, image-to-3D generation, and robotic object manipulation — rather than large, background-heavy scenes. Within this scope we keep the query-based unitary-Gaussian decoder of LeanGaussian and UniGS, but we replace the assumption of known geometry with a geometry transformer (VGGT [26]) that is trained *jointly* with the decoder and predicts per-image cameras and depth. The decoder is then grounded in this learned geometry in three ways: the predicted depth seeds a depth-guided parameterization of each Gaussian centre; fused depth-and-appearance features serve as keys and values for the multi-view deformable attention; and an uncertainty-weighted depth-consistency loss aligns the rendered geometry with the transformer's depth. Because every Gaussian aggregates evidence from all views and the per-view contributions are reconciled in world space, the decoder forms a cross-view consensus that is tolerant to the imperfect poses and depth it is given — a property we exploit rather than assume away. The Gaussian budget is decoupled from resolution and not dictated by the number of views.

We make the following contributions:

- We present FlexSplat, a feed-forward framework that brings query-based, correspondence-free Gaussian reconstruction to the *uncalibrated* object-level setting by training a geometry transformer jointly with the Gaussian decoder, removing the posed-input requirement of prior query-based methods [29, 30].

- We introduce a geometry-grounded decoder: a depth-guided Gaussian parameterization and a multi-view deformable cross-attention over fused depth/appearance features, supervised by an uncertainty-weighted depth-consistency objective. The resulting cross-view consensus is empirically robust to the noise of jointly estimated cameras and depth.

- We show that FlexSplat attains a compact Gaussian representation whose budget is decoupled from input resolution and not dictated by the number of views, that matches posed state-of-the-art reconstructors (*e.g.* UniGS) and matches the best perceptual quality on GSO *without* any camera or depth ground truth.

2 Related Work

Query-based, correspondence-free Gaussian reconstruction. The closest line of work to ours abandons pixel/point correspondence and treats 3D Gaussians as transformer queries. LeanGaussian [29] defines each query as a Gaussian ellipsoid whose centre is a 3D reference point projected onto image features for deformable attention, and refines the queries in a single-view decoder. UniGS [30] generalises this to multiple *posed* views with a multi-view deformable cross-attention (MVDFA) that updates one unitary set of Gaussians shared across views, reducing the "ghosting" and redundancy of per-pixel methods. C3G [1] pushes the same query-based idea towards extreme compactness, reconstructing scenes from unposed images with only $\sim$2K Gaussians. GeoLRM [38] likewise uses deformable cross-attention from 3D anchor points to image features. FlexSplat inherits the decoder design of this family but differs in what it assumes about geometry: rather than consuming given poses (UniGS, GeoLRM) or operating purely in image space, it integrates a *jointly trained* geometry transformer that supplies cameras, depth, and depth-aware features, enabling calibration-free object reconstruction.

Feed-forward object reconstruction. Large reconstruction models predict 3D assets from one or a few images. Splatter Image [23] and its hierarchical extension [20] regress pixel-aligned Gaussians; LGM [24], GRM [33], GS-LRM [39], and MVGamba [36] reconstruct objects from posed multi-view images; InstantMesh [32] and DreamGaussian [25] target mesh/3D content generation. All of these require posed inputs and/or multi-view diffusion to synthesise the views, and most scale their primitive count with the input. FlexSplat instead operates on *uncalibrated* captures and keeps a compact Gaussian budget.

Pose-free and scene-level synthesis. A parallel and very active line targets large, background-heavy *scenes* from unposed images. AnySplat [11] distils geometry priors from VGGT [26] and prunes Gaussians by voxelization; NoPoSplat [35] predicts Gaussians in a canonical frame from unposed pairs; DepthSplat [31] couples Gaussian prediction with monocular depth; MVSplat [4] and PF3plat [9] build cost volumes or use depth/correspondence priors; and WorldMirror [14] unifies many geometric outputs under flexible prior prompting. These methods are trained and evaluated on scene benchmarks (*e.g.* RealEstate10K, DL3DV, ACID) whose imagery, scale, and pixel-aligned representations differ fundamentally from the object-centric, masked-object setting we study; their public models target backgrounds and large baselines rather than single objects. We therefore discuss them as context but do

not claim head-to-head comparison on object benchmarks, which would not be a like-for-like evaluation. The same observation has been raised for object datasets such as ShapeNet-SRN and GSO. FlexSplat occupies the object-level, uncalibrated niche between posed object reconstructors (UniGS, LGM, GRM) and scene-level pose-free models (AnySplat, NoPoSplat).

Geometry foundation models and deformable attention. DUSt3R [28], Fast3R [34], and VGGT [26] predict dense geometry and cameras directly from images and have become strong priors for downstream 3D tasks. We adopt VGGT as the geometry front-end but, unlike methods that consume it frozen as a fixed prior, we train it jointly with the decoder under the rendering and depth-consistency objectives. Deformable attention [41] originates in 2D detection, where features are sampled around reference points; we use its multi-view 3D extension as in [29, 30].

3 Method

FlexSplat maps a set of uncalibrated object images to a single, compact set of 3D Gaussians and renders them to novel views. We review 3D-GS (Sec. 3.1), state the problem (Sec. 3.2), describe the architecture (Sec. 3.3), analyse the compact representation (Sec. 3.4), and give the training objective (Sec. 3.5). The overall pipeline is shown in Figure 1.

3.1 Background: 3D Gaussian Splatting

A scene is represented by anisotropic Gaussian ellipsoids, each with a mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma} = \mathbf{RSS}^\top\mathbf{R}^\top$, decomposed into a quaternion rotation $\mathbf{R}$ and a diagonal scale $\mathbf{S}$ for numerical stability [12]. Each Gaussian carries an opacity $\sigma \in [0,1]$ and spherical-harmonic colour. As an extension of point-based splatting [43], the pixel colour is obtained by alpha-blending the projected Gaussians along the ray, which is differentiable and renders in real time.

3.2 Problem Formulation

Given N RGB images $\{I_i \in \mathbb{R}^{H\times W\times 3}\}_{i=1}^N$ of an object *without* known calibration, we learn a function that predicts (i) per-image geometry — camera parameters $\mathbf{c}_i = \{\mathbf{R}_i, \mathbf{t}_i\}$ and a dense depth map $d_i \in \mathbb{R}^{H\times W}$ — and (ii) a scene representation $\mathbf{G} = \mathcal{G}_\Phi(\{I_i\}_{i=1}^N) = \{g_n\}_{n=1}^{N_G}$ of N_G Gaussians, each $g_n = \{\boldsymbol{\mu}_n, \mathbf{R}_n, \mathbf{S}_n, \sigma_n, \mathbf{SH}_n\}$. A novel view at camera $\pi_{j'}$ is rendered by the differentiable splatter $\mathcal{R}$:

$$I_{j'}^{\text{pred}} = \mathcal{R}\big(\mathcal{G}_\Phi(\{I_i\}_{i=1}^N), \pi_{j'}\big), \tag{1}$$

and supervised against the ground-truth image $I_{j'}$. Crucially, all of $\mathbf{c}_i$, d_i, and $\pi_{j'}$ are *predicted*, not given.

Depth-guided mean parameterization. Regressing absolute centres $\boldsymbol{\mu}_n$ directly is unstable. Following the depth grounding of [23], we tie each centre to the predicted depth and a learnable offset. Let (u_{1i}, u_{2i}) denote the calibrated (normalised) image coordinates of the

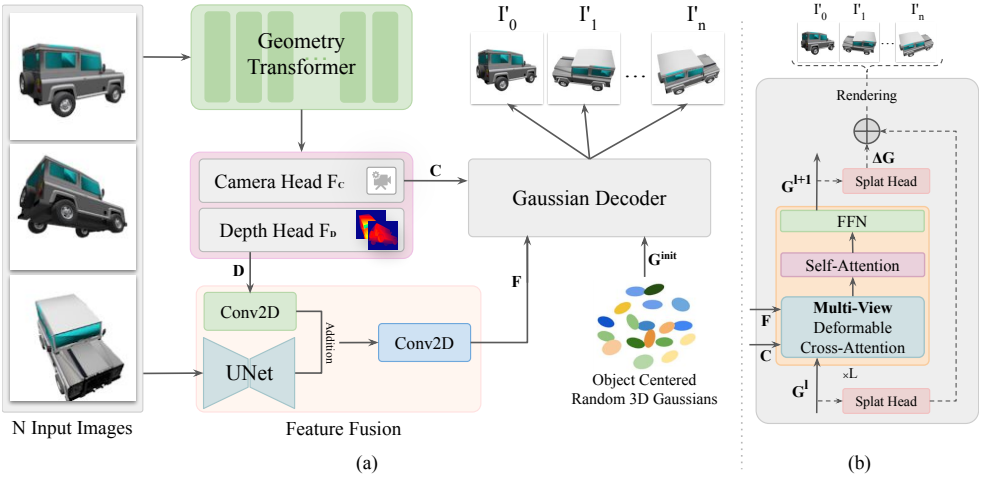

Figure 1: **Overview of FlexSplat.** (a) Given N ($N \geq 1$) uncalibrated input images $\{I_i\}_{i=1}^{N}$, a pretrained VGGT [26], trained jointly with the decoder, predicts camera parameters and depth maps via the camera head F_C and depth head F_D. The predicted depth maps are fused with UNet-extracted pixel-aligned features to obtain depth-aware feature maps F. A set of random 3D Gaussians G^{init}, initialized around the scene center, serves as queries for the Gaussian decoder. (b) The decoder, with L iterative layers, refines Gaussians through multi-view deformable cross-attention, self-attention, and a feed-forward network (FFN). Reference points are projected onto multi-view feature planes at each layer, while the Splat Head renders outputs for supervision. G^l denotes the Gaussian queries at layer l. ⇢ indicates steps used during training only.

corresponding location in view i, and $(\Delta_{x_i}, \Delta_{y_i}, \Delta_{z_i})$ a learnable offset; the centre expressed in the camera frame of view i is the back-projection

$$\boldsymbol{\mu}_{n,i} = \left[u_{1i} d_i + \Delta_{x_i},\ u_{2i} d_i + \Delta_{y_i},\ d_i + \Delta_{z_i} \right]^\top. \tag{2}$$

Grounding the centre in the predicted depth stabilises convergence while leaving room for cross-view refinement; because the depth itself is trained (Sec. 3.5), this prior adapts to the reconstruction task rather than fixing errors in place.

3.3 Architecture

Jointly trained geometry transformer. We use VGGT [26] with two heads: a dense-prediction head [17] producing per-pixel depth and geometry-aware features, and a camera head regressing intrinsics and extrinsics for each image. In contrast to prior pipelines that treat such a model as a fixed front-end, we fine-tune VGGT jointly with the decoder, so its depth and camera predictions are shaped by the rendering and depth-consistency objectives. This is the property that lets the system partially self-correct: the geometry is not asked to be right in isolation, only to be useful for the reconstruction it is supervised to produce.

Dual-branch features and fusion. A UNet encoder–decoder [18] extracts pixel-aligned appearance features in parallel with the geometry branch. Following the FPN-style fusion

of [13, 29], depth features are added to the UNet features and refined by a convolution, then reduced by a lightweight convolution for efficiency. The fused multi-view features F serve as the keys and values of the deformable cross-attention. A reconstruction loss on the UNet features enforces pixel-wise consistency between input and reconstructed views.

Geometric initialization. Random Gaussian placement is unstable in the multi-view setting, where sampling a projected centre that lands off-image yields zero features and vanishing gradients. We therefore place all queries near the estimated scene centre, computed as the least-squares intersection of the camera optical axes from the camera head, and regress the initial parameters $\mathbf{G}_{\text{init}} = \mathcal{S}(\mathbf{q}_{\text{init}})$ with a splatting head using Xavier-uniform weights and fixed attribute biases.

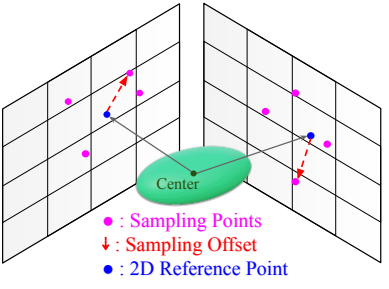

Figure 2: **Multi-view projection of Gaussian centers.** Each 3D Gaussian is projected onto multiple image planes, where its 2D reference points and learnable offsets define sampling locations for multi-view deformable attention.

Multi-view deformable cross-attention. We adopt the multi-view deformable formulation of [29, 30]. Each query is a Gaussian ellipsoid with no native 2D location; we obtain one by projecting its 3D centre into every view (Figure 2). Let $\pi_i(\cdot)$ denote the perspective projection into view i (the rigid transform by $(\mathbf{R}_i, \mathbf{t}_i)$, the intrinsics $\mathbf{K}_i$, and the division by depth that maps to pixel coordinates). At decoder layer l, the centre $\boldsymbol{\mu}_n^l$ yields N reference points $\{\pi_1(\boldsymbol{\mu}_n^l), \ldots, \pi_N(\boldsymbol{\mu}_n^l)\}$ describing how the same primitive is seen across views. Around each reference point we sample N_P locations with learned offsets and aggregate them with attention weights. Figure 2 illustrates the 2D reference point, the sampled locations, and their offsets in each view. The aggregation is:

$$\mathbf{q}_n^{l'} = \sum_{i=1}^{N} \sum_{p=1}^{N_P} A_{n,i,p}^l \,\text{Bilinear}\big(F_i,\ \pi_i(\boldsymbol{\mu}_n^l) + \Delta\mathbf{s}_{n,i,p}^l\big), \tag{3}$$

where $\Delta\mathbf{s}_{n,i,p}^l = \text{MLP}_{\text{off}}(\mathbf{q}_n^l)$ are the learned sampling offsets and the weights $A_{n,i,p}^l = \text{softmax}(\text{MLP}^A(\mathbf{q}_n^l))$ are normalised jointly over all views and sampling points ($\sum_{i,p} A_{n,i,p}^l = 1$). Because a single query attends jointly across all N views and the resulting update is reconciled in world space, each Gaussian forms a cross-view consensus rather than committing to any one (possibly noisy) projection — the mechanism that absorbs moderate camera and depth error.

Gaussian decoder and refinement. We learn N latent queries $\mathbf{q} \in \mathbb{R}^{N\times C}$ refined over L layers. Each layer applies multi-view deformable cross-attention (Eq. 3), self-attention across queries, and an FFN:

$$\mathbf{q}^{l+1} = \text{FFN}\big(\text{SelfAttn}(\text{MV-DeformAttn}(\mathbf{q}^l, \mathbf{F}, \boldsymbol{\mu}^l))\big), \tag{4}$$

and a splatting head predicts incremental updates $\Delta\boldsymbol{G}^l = \mathcal{S}^l(\mathbf{q}^l)$ that are composed onto the running estimate ($\mathbf{G}^{l+1} = \mathbf{G}^l \oplus \Delta\boldsymbol{G}^l$: addition for translation/appearance, multiplica-

tion for rotation). After each layer the refined centres are re-projected to update the reference points, and each layer's Gaussians are rendered and supervised, giving progressive, geometry-consistent refinement.

3.4 A Compact, Resolution-Independent Representation

A direct consequence of the query-based design is that the number of primitives N_G is a free hyper-parameter, set independently of the input rather than dictated by it. This contrasts sharply with pixel-aligned reconstructors, whose primitive count is $\Theta(H \cdot W \cdot N)$ and therefore grows with both resolution and the number of views. For four input views at 256×256, a pixel-aligned method instantiates on the order of $256^2 \times 4 \approx 2.6 \times 10^5$ Gaussians, whereas FlexSplat uses 2×10^4 — about an order of magnitude fewer — and, crucially, this budget is independent of image resolution. The budget is also not tied to the number of views: we set it to 10K/15K/20K for 1/2/4 views (Sec. 4.6), a modest scaling chosen to give more views more capacity, though it could equally be held fixed. Even at the largest setting the primitive count stays roughly an order of magnitude below a pixel-aligned four-view budget. The same motivation drives recent compact query-based work such as C3G [1].

3.5 Training Objective

With predicted depth and per-pixel confidence $(D_i^{\mathrm{VGGT}}, \mathcal{U}_i)$ from the geometry head, the loss combines an RGB reconstruction term on the feature-extractor output $\hat{I}_i^E$ and every decoder layer $\hat{I}_{i,l}^D$, a confidence-weighted depth-consistency term aligning the rendered depth $\hat{D}_i$ with D_i^{VGGT} (the weighting down-scales pixels the geometry head deems unreliable), and an LPIPS [40] perceptual term:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\Big(\lambda_E\,\mathcal{L}_{\mathrm{MSE}}(\hat{I}_i^E, I_i) + \sum_{l=1}^{L}\lambda_D\,\mathcal{L}_{\mathrm{MSE}}(\hat{I}_{i,l}^D, I_i)\Big) + \frac{\lambda_{\mathrm{Depth}}}{N}\sum_{i=1}^{N}\big\|(\hat{D}_i - D_i^{\mathrm{VGGT}})\odot\mathcal{U}_i\big\|_1 + \frac{1}{N}\sum_{i=1}^{N}\sum_{l=1}^{L}\lambda_{\mathrm{LPIPS}}\,\mathcal{L}_{\mathrm{LPIPS}}(\hat{I}_{i,l}^D, I_i). \tag{5}$$

Because the geometry transformer is in the gradient path, the depth-consistency term both regularises the Gaussians and adapts the predicted geometry to the rendering task.

4 Experiments

4.1 Implementation details

Following [23], each Gaussian is parameterised by 24 values: 4 for the mean (1 depth, 3 offsets), 7 for the covariance (3 scale, 4 rotation), 1 opacity, and 12 spherical-harmonic colour coefficients. The decoder uses L=3 layers, trading off memory, accuracy, and speed. We use 10K Gaussians for single-view and 20K for four-view testing for budget parity with baselines. Training uses Adam with learning rate 1×10^{-4} on 4× RTX 4090 (24 GB) GPUs, and begins with RGB MSE and depth losses, adding LPIPS later for perceptual quality.

Table 1: Single-view NVS on ShapeNet-SRN, averaged per category. Best per column in **bold**.

Method	Chairs			Cars		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
SRN [21]	22.89	0.890	0.104	22.25	0.880	0.129
CodeNeRF [10]	23.66	0.900	0.166	23.80	0.910	0.128
ViewsetDiff (w/o depth) [23]	24.16	0.910	0.088	23.21	0.900	0.116
PixelNeRF [32]	23.72	0.900	0.128	23.17	0.890	0.146
NeRFDiff (w/o NGD) [4]	24.80	0.930	0.070	23.95	0.920	0.092
Splatter Image [25]	24.43	0.930	0.067	24.00	0.920	0.078
Hierarchical SI [20]	25.43	0.940	0.066	24.18	0.920	0.087
LeanGaussian [29]	**25.87**	**0.950**	**0.065**	25.00	**0.930**	**0.075**
FlexSplat (Ours)	25.48	**0.950**	0.066	**25.09**	0.920	0.079

4.2 Datasets and metrics

For single-view reconstruction we use ShapeNet-SRN [21] (Cars, Chairs). For multi-view training we use a subset of Objaverse-LVIS [8] (>1,000 categories), sampling 1–8 views per instance, and evaluate on 250 objects from Google Scanned Objects (GSO) [6] with four input views. We report PSNR, SSIM, and LPIPS between rendered and held-out target views. FlexSplat uses no camera poses *at input*; for quantitative evaluation, the cameras predicted by the geometry head are aligned to the evaluation coordinate frame by a global similarity transform, so that each held-out target view can be rendered and compared against its ground truth, as is standard for pose-free NVS [35].

4.3 Single-view reconstruction on ShapeNet-SRN

Table 1 compares FlexSplat with single-view reconstructors on ShapeNet-SRN. FlexSplat is on par with the strongest query-based baseline, LeanGaussian [29] — slightly ahead on Cars PSNR and within 0.4 dB on Chairs — while, unlike LeanGaussian, it does not assume the input view is the reference frame and instead infers the camera. Qualitatively (Figure 3), FlexSplat recovers sharper boundaries and thin structures, indicating that the multi-view aggregation benefits even the single-view regime.

4.4 Multi-view reconstruction on GSO

Table 2 reports four-view GSO results and marks which methods require posed inputs. Every baseline is posed; FlexSplat is the only pose-free method, and like the others it uses no ground-truth depth (its depth is predicted by the geometry head). Under this handicap FlexSplat rivals the posed state of the art: it is within 0.7 dB PSNR and 0.012 SSIM of UniGS [30] — the method whose query-based decoder is closest to ours but which is given the cameras — and it matches the best LPIPS among the compared methods (0.041 vs. 0.042 for UniGS, within noise). We read this as evidence that calibration-free operation incurs only a small fidelity cost here (0.69 dB PSNR and 0.012 SSIM below UniGS, with comparable LPIPS) while removing UniGS's posed-input assumption. Qualitatively (Figure 4), FlexSplat avoids the duplicate "ghost" Gaussians of LGM [24] and the missing fine structures of InstantMesh [32], and is visually comparable to UniGS despite using no calibration.

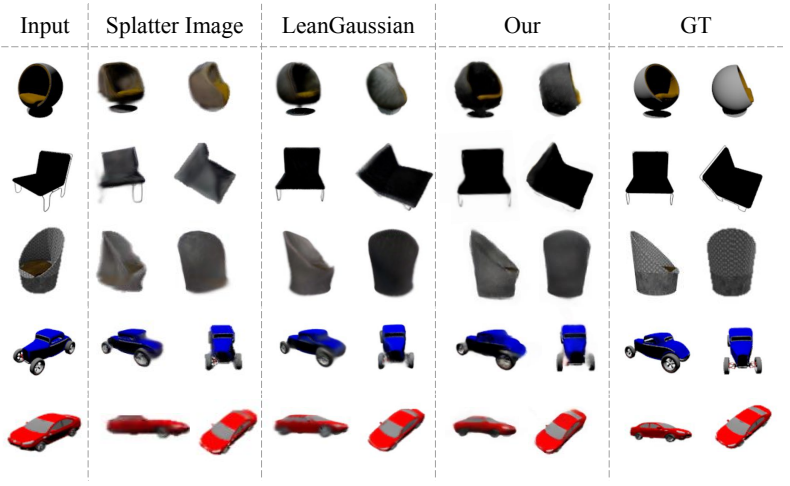

Figure 3: Single-view NVS on ShapeNet-SRN. FlexSplat improves geometric consistency and structural detail over Splatter Image [23] and LeanGaussian [29], with comparable colour fidelity.

Table 2: Four-view NVS on GSO. "Posed" marks methods that require ground-truth or input camera poses; FlexSplat is the only pose-free method. No listed method requires ground-truth depth. †Splatter Image uses cameras only inside the renderer, not as network input. "NA": not reported by the source. Best per metric in **bold**.

Method	GSO (4 views)			Posed
	PSNR↑	SSIM↑	LPIPS↓	
Splatter Image [23]	25.62	0.915	0.152	✓†
LGM (Small) [24]	17.48	0.783	0.218	✓
LGM (Large) [24]	26.25	0.925	0.054	✓
InstantMesh [32]	23.02	0.889	0.089	✓
GeoLRM [38]	22.84	0.851	NA	✓
MVGamba [36]	26.25	0.881	0.069	✓
GRM (Res-512) [33]	30.05	0.906	0.052	✓
GS-LRM (Res-256) [39]	29.59	0.944	0.051	✓
UniGS [30]	**30.42**	**0.961**	0.042	✓
FlexSplat (Ours)	29.73	0.949	**0.041**	✗

4.5 Single-view reconstruction on GSO

Table 3 reports single-view GSO results following the UniGS protocol (views for non-native-single-view baselines generated with ImageDream [27]). FlexSplat exceeds LGM and InstantMesh in PSNR and matches the best LPIPS, remaining competitive with UniGS in this open-domain single-view setting as well.

4.6 Effect of the number of input views

Table 4 and Figure 5 quantify the effect of the number of input views. Going from one to two views yields the largest gain (+5.17 dB PSNR), as the second view resolves most of the single-view ambiguity; a fourth view adds a further +2.63 dB, with diminishing returns.

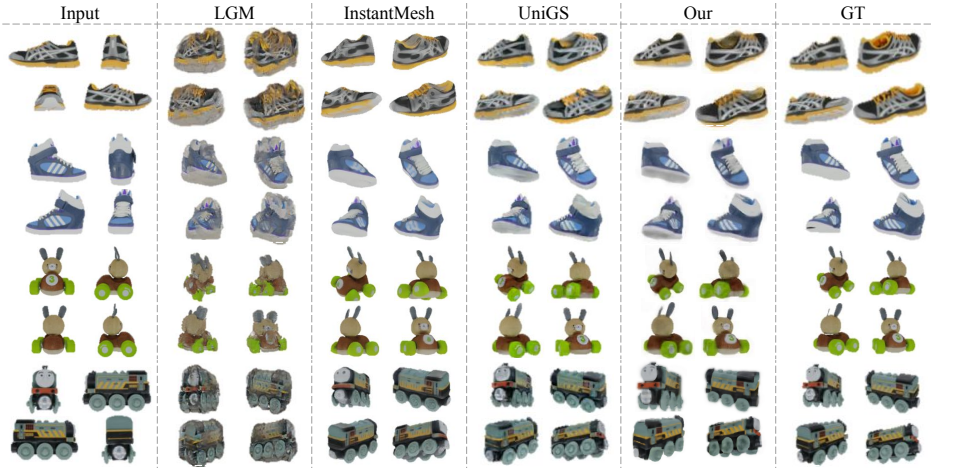

Figure 4: Four-view NVS on GSO. FlexSplat produces consistent geometry and texture without any known camera parameters.

Table 3: Single-view reconstruction on GSO. Best per metric in **bold**.

Method	PSNR↑	SSIM↑	LPIPS↓
LGM [24]	20.81	**0.858**	0.151
InstantMesh [33]	19.47	0.838	0.184
UniGS [30]	**22.35**	0.857	**0.149**
FlexSplat (Ours)	21.93	0.850	**0.149**

Table 4: Effect of the number of input views. The same 250 GSO objects and identical held-out target views are used across all view counts; only the number of input views varies. The one- and four-view rows correspond to Tables 3 and 2. Best per metric in **bold**.

Input views (budget)	PSNR↑	SSIM↑	LPIPS↓
1 view (10K)	21.93	0.850	0.149
2 views (15K)	27.10	0.921	0.072
4 views (20K)	**29.73**	**0.949**	**0.041**

We scale the budget modestly with views (10K/15K/20K for 1/2/4 views) to match the additional surface coverage; because the number of Gaussian queries is a free hyper-parameter independent of the view count, no retraining is needed to change the input count.

4.7 Inference time

Table 5 reports timings. FlexSplat adds a small overhead over LeanGaussian and UniGS because it predicts cameras and depth rather than receiving them, but remains real-time: a single view in 0.706 s and four views in 0.992 s end-to-end, faster than most alternatives. The overhead is the price of removing calibration, and it is modest.

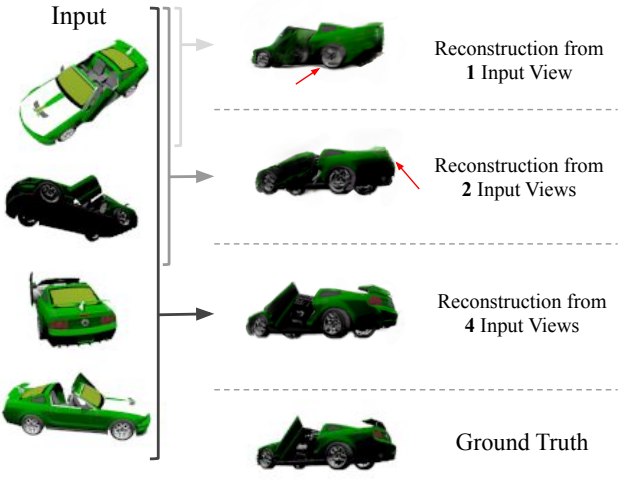

Figure 5: Effect of the number of input views (1, 2, 4) for the same object. More views give the unitary representation more cross-view evidence, improving reconstruction. Viewpoints are randomly sampled.

Table 5: Inference time (seconds). 3D: reconstruction; R: render one view; Inference $\approx$ 3D $+n \cdot R$ (one forward pass plus n novel-view renders, n=250 single-view and 32 multi-view). ‡PixelNeRF timings are quoted from prior work and not recomputed.

	Method	3D↓	R↓	Inference↓
Single-view	PixelNeRF [63]‡	0.0050	1.2200	304.5300
	OpenLRM [8]	2.2400	2.1200	532.2400
	Splatter Image [73]	0.0220	0.0025	0.6470
	Triplane Gaussian [82]	1.2810	0.0025	1.9060
	LeanGaussian [29]	0.1400	0.0018	0.5900
	FlexSplat (Ours)	0.1810	0.0021	0.7060
Multi-view	DreamGaussian [75]	118.3245	0.0038	118.4461
	InstantMesh [62]	0.6049	0.6206	20.4641
	LGM [74]	1.6263	0.0090	1.9143
	UniGS [80]	0.6939	0.0019	0.7538
	FlexSplat (Ours)	0.9023	0.0028	0.9919

4.8 Ablation study

Table 6 ablates the main components on the Objaverse-LVIS validation split (its absolute numbers differ from the GSO test set in Table 2 because it is a distinct, held-out split). Most important for our central claim, freezing VGGT rather than training it jointly with the decoder costs 1.38 dB PSNR (26.28 $\rightarrow$ 24.90), 0.024 SSIM, and 0.013 LPIPS, isolating the benefit of co-adapting the geometry front-end — the design choice that separates FlexSplat from posed query-based methods that consume a fixed geometry estimate. Removing geometric initialization (random centres) collapses training: projected centres land off-image, features vanish, and gradients die. Removing the VGGT depth features sharply reduces PSNR/SSIM and raises LPIPS, confirming that depth-aware features are central to

Table 6: Ablation on Objaverse-LVIS validation. Best per metric in **bold**.

Variant	PSNR↑	SSIM↑	LPIPS↓
Random initialization	11.94	0.632	0.712
w/o depth-consistency loss	25.99	0.902	0.073
w/o VGGT depth features	23.52	0.822	0.095
Frozen VGGT (no joint training)	24.90	0.896	0.078
Frame-wise Gaussians	24.11	0.833	0.086
Full model	**26.28**	**0.920**	**0.065**

stabilising multi-view reasoning. Removing the depth-consistency term degrades geometric alignment more mildly, indicating it acts as a useful regulariser. Replacing the shared unitary Gaussian set with frame-wise independent Gaussians causes a clear drop, underscoring the value of a single cross-view-consistent representation. These ablations also speak to robustness: the components that ground the decoder in predicted geometry (depth features, depth consistency) are precisely the ones that let the system tolerate the imperfect cameras and depth it estimates.

4.9 Robustness to camera-pose perturbation

Table 7 probes how much pose error the decoder can absorb. Starting from the cameras estimated by the geometry head on GSO (four views), we add independent random rotation noise of magnitude σ to each camera's orientation before it is used to project Gaussians into that view for deformable attention, and evaluate the reconstruction against the clean held-out target views; metrics are averaged over the 250 GSO test objects. For reference, the geometry head's own predicted cameras deviate from ground truth by 3° in rotation and 3% of object scale in camera-center position (250 objects, after the similarity alignment of Sec. 4.2); the headline 29.73 dB (Table 2) is reached at this operating point, so FlexSplat comes within 0.69 dB of posed UniGS *despite* imperfect estimated poses. Degradation is graceful: a 1° perturbation costs only 0.22 dB PSNR, and even at 5° — a substantial orientation error — PSNR remains at 27.64 with LPIPS 0.061. Quality falls off faster only for large errors ($\geq$ 10°), where misaligned views can no longer be reconciled and four-view fusion approaches the quality of a single clean view (see Table 3). This behaviour is consistent with the cross-view consensus of Sec. 3.3: because each Gaussian aggregates evidence over all views in world space, moderate per-view pose error tends to be averaged down rather than propagated. We perturb rotation only; a full translational sensitivity analysis is left to future work.

Table 7: Robustness to camera-pose perturbation (GSO, four views). Independent random rotation noise of magnitude σ is added to the estimated camera orientations at inference; metrics are averaged over the 250 GSO test objects. ΔPSNR is relative to the previous row.

Rotation noise σ	PSNR↑	SSIM↑	LPIPS↓	ΔPSNR
0° (no noise)	29.73	0.949	0.041	–
1°	29.51	0.946	0.043	−0.22
2°	28.96	0.941	0.047	−0.55
5°	27.64	0.928	0.061	−1.32
10°	25.12	0.896	0.094	−2.52
20°	21.08	0.832	0.151	−4.04

5 Conclusion and Limitations

We presented FlexSplat, a feed-forward framework that brings query-based, correspondence-free Gaussian reconstruction to the *uncalibrated*, object-level setting. By training a geometry transformer jointly with a multi-view deformable decoder and grounding the Gaussians in predicted depth, FlexSplat produces a compact, resolution-independent representation and rivals posed state-of-the-art reconstructors on GSO without using any camera or depth ground truth, while matching the best perceptual quality among the compared methods.

Limitations. FlexSplat is trained and evaluated on object-centric captures; extending the same calibration-free decoder to large, background-heavy scenes (in the manner of scene-level pose-free models) is left to future work and would require scene-scale training data and benchmarks. Although the geometry transformer is trained jointly and the cross-view averaging absorbs moderate pose and depth error, very large geometric errors — *e.g*. under dense or highly inconsistent captures — can still propagate into the reconstruction. Finally, our gains over the strongest posed baseline are in perceptual quality and in the removal of the calibration requirement rather than in raw PSNR; while our pose-perturbation study (Sec. 4.9) shows graceful degradation under moderate rotation error, a full translational sensitivity analysis remains future work.

References

[1] Honggyu An, Jaewoo Jung, Mungyeom Kim, Sunghwan Hong, Chaehyun Kim, Kazumi Fukuda, Minkyeong Jeon, Jisang Han, Takuya Narihira, Hyuna Ko, Junsu Kim, Yuki Mitsufuji, and Seungryong Kim. C3G: Learning compact 3D representations with 2K gaussians. *arXiv preprint arXiv:2512.04021*, 2025.

[2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022.

[3] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D gaussian splats from image pairs for scalable generalizable 3D reconstruction. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2024.

[4] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: Efficient 3D gaussian splatting from sparse multi-view images. In *Eur. Conf. Comput. Vis. (ECCV)*, 2024.

[5] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A dataset of annotated 3D objects in the wild. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2023.

[6] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3D scanned household items. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2022.

[7] Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. NerfDiff: Single-image view synthesis with NeRF-guided distillation from 3D-aware diffusion. In *Int. Conf. Machine Learning (ICML)*, 2023.

[8] Zexin He and Tengfei Wang. OpenLRM: Open-source large reconstruction models. https://github.com/3DTopia/OpenLRM, 2024.

[9] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jisang Han, Jiaolong Yang, Chong Luo, and Seungryong Kim. PF3plat: Pose-free feed-forward 3D gaussian splatting. *arXiv preprint arXiv:2410.22128*, 2024.

[10] Wonbong Jang and Lourdes Agapito. CodeNeRF: Disentangled neural radiance fields for object categories. In *Int. Conf. Comput. Vis. (ICCV)*, 2021.

[11] Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. AnySplat: Feed-forward 3D gaussian splatting from unconstrained views. *ACM Trans. Graph. (TOG)*, 44(6), 2025.

[12] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph. (TOG)*, 42(4), 2023.

[13] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017.

[14] Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. WorldMirror: Universal 3D world reconstruction with any-prior prompting. *arXiv preprint arXiv:2510.10726*, 2025.

[15] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Eur. Conf. Comput. Vis. (ECCV)*, 2020.

[16] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph. (TOG)*, 41(4), 2022.

[17] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Int. Conf. Comput. Vis. (ICCV)*, 2021.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Int. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[19] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.

[20] Jianghao Shen, Nan Xue, and Tianfu Wu. A pixel is worth more than one 3D gaussians in single-view 3D reconstruction. *arXiv preprint arXiv:2405.20310*, 2024.

[21] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2019.

[22] Stanisław Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Viewset diffusion: (0-)image-conditioned 3D generative models from 2D data. In *Int. Conf. Comput. Vis. (ICCV)*, 2023.

[23] Stanisław Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3D reconstruction. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2024.

[24] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. LGM: Large multi-view gaussian model for high-resolution 3D content creation. In *Eur. Conf. Comput. Vis. (ECCV)*, 2024.

[25] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. DreamGaussian: Generative gaussian splatting for efficient 3D content creation. *Int. Conf. Learn. Represent. (ICLR)*, 2024.

[26] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2025.

[27] Peng Wang and Yichun Shi. ImageDream: Image-prompt multi-view diffusion for 3D generation. *arXiv preprint arXiv:2312.02201*, 2023.

[28] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2024.

[29] Jiamin Wu, Kenkun Liu, Han Gao, Xiaoke Jiang, and Lei Zhang. LeanGaussian: Breaking pixel or point cloud correspondence in modeling 3D gaussians. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2025.

[30] Jiamin Wu, Kenkun Liu, Yukai Shi, Xiaoke Jiang, Yuan Yao, and Lei Zhang. UniGS: Modeling unitary 3D gaussians for novel view synthesis from sparse-view images. In *Int. Conf. Comput. Vis. (ICCV)*, 2025.

[31] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. DepthSplat: Connecting gaussian splatting and depth. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2025.

[32] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. InstantMesh: Efficient 3D mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.

[33] Yinghao Xu, Zifan Shi, Yifan Wang, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. GRM: Large gaussian reconstruction model for efficient 3D reconstruction and generation. In *Eur. Conf. Comput. Vis. (ECCV)*, 2024.

[34] Jianing Yang, Alexander Sax, Kevin J. Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3D reconstruction of 1000+ images in one forward pass. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2025.

[35] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3D gaussian splats from sparse unposed images. In *Int. Conf. Learn. Represent. (ICLR)*, 2025.

[36] Xuanyu Yi, Zike Wu, Qiuhong Shen, Qingshan Xu, Pan Zhou, Joo-Hwee Lim, Shuicheng Yan, Xinchao Wang, and Hanwang Zhang. MVGamba: Unify 3D content generation as state space sequence modeling. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2024.

[37] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2021.

[38] Chubin Zhang, Hongliang Song, Yi Wei, Yu Chen, Jiwen Lu, and Yansong Tang. GeoLRM: Geometry-aware large reconstruction model for high-quality 3D gaussian generation. In *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2024.

[39] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. GS-LRM: Large reconstruction model for 3D gaussian splatting. In *Eur. Conf. Comput. Vis. (ECCV)*, 2024.

[40] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.

[41] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *Int. Conf. Learn. Represent. (ICLR)*, 2021.

[42] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3D reconstruction with transformers. *arXiv preprint arXiv:2312.09147*, 2023.

[43] Matthias Zwicker, Hanspeter Pfister, Jeroen van Baar, and Markus Gross. EWA volume splatting. In *IEEE Visualization (VIS)*, pages 29–36, 2001.